In the Specification:

Please amend the paragraph beginning at page 7, line 26 as follows:

According to the preferred deadman circuitry **54a** of FIG. 4A, which is also referred to herein as a type of dead man switch, a first data stream P may be generated by a weak random data generator **70**. As illustrated, the weak random data generator **70** may provide a weak random data stream in response to a clock signal and a noise signal and may be of conventional design. An outgoing data framer **72** and an open-drain driver **78** may also be provided so that the first data stream P can be passed to an input/output pad (I/O). This first data stream P may be provided in-sync with timing signals generated by a central timing circuit **74**. Other formats for the first data stream P may also be used.

Please amend the paragraph beginning at page 11, line 1 as follows:

The second encrypted data stream R is preferably generated by performing an encryption operation that evaluates the first data stream P and a plurality of previously generated bits in the second encrypted data stream R. As illustrated by the flow diagram of operations shown in FIG. 4C, the second stream encryptor **100** within the authorization device **56** may use conventional permuting operations to sequentially determine a plurality of permuted bits as $\{H_1, H_2, H_3, \ldots, H_n\}$ during a first time interval, with each permuted bit being determined in accordance with the following expression:

$$H_i = f_p (P_i, R_{i-j}, \ldots R_{i-j-k})$$

where f_p is a permuting function, "i" and "j" are positive integers and "k" represents a preferred "depth" to which the first data stream R is evaluated. Accordingly, each of these permuted bits $H_1, H_2, \ldots, H_n$ is generated at a respective point in the first time interval. The second stream encryptor **100** within the authorization device **56** may also use a conventional encryption key (f_{key}) to generate the second encrypted data stream from the permuted bits in accordance with the following expression:

$$R_{i+1} = f_{key} (H_i, H_{i-l}, \ldots H_{i-l-m})$$

where “l” and “m” are positive integers. Other conventional permuting operations and encryption keys may also be used and those described herein are provided as exemplary operations for generating an encrypted data stream.